REMARKS

Claims 1-2, 4, and 7-15 are pending in the application, with claims 1, 7, 10, and 13 being independent. Applicant has amended claim 8. No new matter has been added by way of the foregoing amendments.

The Examiner rejected claims 1 and 7-15 under 35 U.S.C. 103(a) as being unpatentable over Saulsbury et al., U.S. Patent No. 6,314,510 (hereinafter referred to as Saulsbury). The examiner also rejected claims 1, 7, 10 and 13 under 35 U.S.C. §102(e) as being anticipated by Steven's "ALU Design and Processor Branch Architecture," Microprocessing and Microprogramming, 1993.

In the rejections, the examiner relies in part on "official notice." MPEP 2104.04 states:

> If applicant adequately traverses the examiner's assertion of official notice, the examiner must provide documentary evidence in the next Office action if the rejection is to be maintained. See 37 CFR 1.104(c)(2). See also Zurko, 258 F.3d at 1386, 59 USPQ2d at 1697 ("[T]he Board [or examiner] must point to some concrete evidence in the record in support of these findings" to satisfy the substantial evidence test). If the examiner is relying on personal knowledge to support the finding of what is known in the art, the examiner must provide an affidavit or declaration setting forth specific factual statements and explanation to support the finding. See 37 CFR 1.104(d)(2).

Applicant, therefore, traverses the examiner's assertion of "official notice" and requests that the examiner provide documentary evidence to support the rejection of claims 1 and 7-15, if these rejections are maintained. Regarding claim 1, the examiner stated:

> Referring to claim 1, Saulsbury has taught a computer program product residing on a computer readable storage medium (Fig.I, component 102; column 2, lines 42-45) comprising instructions (Fig.4), including a context branch instruction (Fig.4 and column 4, lines 44-45; note the "branch on zero" (bz) instruction) that, when executed, causes a data processing apparatus to select another instruction in an instruction stream from one of a branch target instruction associated with a label specified by the context branch instruction (Fig.4; note that the "bz nextl" instruction will branch to the target instruction associated with label "nextl", which is the "btst wrl, dbl" instruction) and an instruction following the context branch instruction ("st WTO, [addr2]") based on a comparison of a current number to a number specified by the context branch instruction. The bz

instruction is a branch on zero instruction, which means that if the number specified by the branch instruction (zero) matches a current number (i.e., the value to be compared to zero), then a branch will occur. In this program, the "bz next!" instruction checks to see if the current number (which may be interpreted as either dirty bit dbO or the flag which would be checked by the bz instruction) is equal to 0, the specified number. See column 4, lines 39-45. It should be noted that the branch is a context branch as the branch is associated with executing contexts. See the title, abstract, and column 2, lines 40-41.
b) retrieving the selected other instruction. As is known, with a condition branch, as shown in Fig.4, either the target instruction or the subsequent instruction will be retrieved.
c) despite teaching multiple contexts, Saulsbury has not explicitly taught that the data processing apparatus is a multi-threaded data processing apparatus and that the current number is a current executing thread number and that the specified number is a thread number specified by the branch instruction. However, Official Notice is taken that multithreading and its advantages are well known and accepted in the art. Specifically, a multithreaded system executes multiple independent groups of instructions called threads. By having multiple independent groups of instructions as opposed to a single serial stream of instructions, a multithreaded system is able to switch to another thread when one thread experiences a delay. This ensures that processor resources are kept as busy as possible, which in turn increases throughput. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Saulsbury's system such that it supported multithreading. It would then follow that the aforementioned branch instruction would be executed on the multithreaded system and the current number would be the current executing thread number (i.e., a number associated with the current thread), and the thread number specified by the branch instruction would still be zero, as described above, because zero is a thread number (i.e., a number associated with a thread). It should be noted that it would be obvious to have this instruction in any multithreaded machine as "branch on zero", as it is a very well-known instruction in the art.

Applicant's independent claim 1 recites "a context branch instruction that, when executed, causes a data processing apparatus to: select another instruction in an instruction stream from one of a branch target instruction associated with a label specified by the context branch instruction and an instruction following the context branch instruction based on a comparison of a current executing thread number to a thread number specified by the context branch instruction; and retrieve the selected other instruction."

Applicant disagrees with the Examiner's assertion that it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Saulsbury with

the broad principles of multithreading. In fact, certain results discussed by the Examiner that could be obtained by combining Saulsbury with the principles of multithreading would not be obvious to one of ordinary skill in the art. For instance, the Examiner states that if Saulsbury's system were modified to support multithreading, "[i]t would then follow that the aforementioned branch instruction would be executed on the multithreaded system and the current number would be the current executing thread number." (Office Action, page 4).

Earlier in the office action, the Examiner also stated that in Saulsbury, the current number may be "either dirty bit db0 or the flag which would be checked by the bz instruction." (Office Action, page 3). However, Saulsbury indicates that "the dirty bits currently stored by all of the dirty bit registers dbr- to dbrN-1 indicate which of the working registers wr0 to wrN-1 store operands that are to be saved to the main memory at the next context switch." (Col. 3, lines 30-33). Thus, Saulsbury describes at least one example in which the current number represented by db0 would not be a current executing thread number, even if the system were modified to support multithreading. The dirty bit registers store information related to the status of data, and it is therefore illogical to assume that the dirty bits would be thread numbers if the system were multithreaded, as the examiner appears to imply. Even if it were obvious to combine multithreading with the system described by Saulsbury (which Applicant does not concede), it would not be obvious to compare thread numbers in the manner recited in Applicant's claim 1.

For similar reasons, the Examiner rejected independent claim 1 as being unpatentable over Steven. Again, the Examiner asserts that if the system described by Steven was modified to be multithreaded, "[i]t would then follow that the aforementioned branch instruction would be executed on the multithreaded system and the current number would be the current executing thread number." (Office Action, page 12). The Examiner further states that "[t]he Bcc instruction compares a register value (specified number) to a current number (either another register value or immediate specified by the branch instruction)." (Id).

Even if it were obvious to combine multithreading with the system described by Saulsbury (which Applicant does not concede), it does not follow that the register values and immediate values described by Steven would be thread numbers. For instance, Steven states:

> Here fully general compare and branch instructions are only provided to test for equality and inequality. The four remaining branch instructions always compare a single operand against zero. (Steven, Page 268, right column).

In this example of Steven, the value zero is used for comparison with an operand. The value zero however is not a thread number in this instance. Nothing in the references teaches or suggests that the information in the registers could be a thread number, and thus it would not be obvious to compare thread numbers in the manner described in Applicant's claim 1, even if the system described by Steven were modified to be multithreaded.

Independent claims 7, 10 and 13 recite "performing a comparison of a thread number of an executing thread to a thread number specified by a context branch instruction; selecting another instruction in an instruction stream from one of a branch target instruction associated with a label specified by the context branch instruction and an instruction following the context branch instruction based on the comparison," or similar language. For reasons similar to those provided with respect to independent claim 1, at least these features are not disclosed by the cited art. Applicant's independent claims 7, 10 and 13, and the claims that respectively depend from them, are therefore patentable over the cited art.

All of the dependent claims are patentable for at least the reasons for which the claims on which they depend are patentable.

In view of the foregoing, applicant respectfully submits that the application is in condition for allowance and such action is respectfully requested at the examiner's earliest convenience.

Canceled claims, if any, have been canceled without prejudice or disclaimer.

Any circumstance in which the applicant has (a) addressed certain comments of the examiner does not mean that the applicant concedes other comments of the examiner, (b) made arguments for the patentability of some claims does not mean that there are not other good reasons for patentability of those claims and other claims, or (c) amended or canceled a claim does not mean that the applicant concedes any of the examiner's positions with respect to that claim or other claims.

Please charge the Petition for Extension of Time fee of **$120** and please apply any other charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: January 22, 2008

/Denis G. Maloney/
Denis G. Maloney
Reg. No. 29,670

Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21813082.doc